

# L'informatica e i linguaggi di programmazione

## Cos'è l'Informatica

---

Informatica = Informazione (dati + istruzioni) + Automazione.

L'informatica è la scienza che studia il computer sotto tutti i suoi aspetti. In inglese si usa l'espressione "computer science", mentre in italiano si preferisce la parola "informatica", che nasce dalla contrazione delle parole *informazione* e *automatica*, per indicare l'elaborazione automatica dell'informazione, che è appunto l'attività effettuata dal computer.

L'informatica è una scienza che non si occupa di oggetti naturali (a differenza di fisica, chimica e biologia) o di oggetti ideali (a differenza della matematica), bensì di oggetti sociali, costruiti dagli uomini secondo un progetto e per degli scopi ben precisi. In tal senso l'informatica assomiglia a discipline come il diritto e l'economia, pertanto l'informatica si può considerare come un sistema di regole che devono essere stabilite e seguite perché l'uso del computer sia efficace.

Per capire come funzionano i computer, occorre capire che cos'è un sistema di regole e che cosa significa seguirle. Le regole sono leggi che vengono stabilite all'interno di una comunità e che le persone accettano di seguire per raggiungere determinati obiettivi. L'informatica è un sistema di regole per la costruzione, il funzionamento e l'uso dei computer.

Soltanto se siamo in grado di seguire le regole sociali possiamo imparare le regole informatiche. Dunque i due momenti sono parte di un'unica valutazione.

## Cos'è il Computer

---

È un insieme organizzato di componenti, in grado di eseguire una successione di istruzioni finalizzate a elaborare l'informazione. È in grado di eseguire operazioni relativamente semplici ad alta velocità.

La differenza fondamentale fra il computer e i suoi antenati (primo fra tutti la calcolatrice) è che il computer è in grado di ricevere e memorizzare non soltanto semplici dati (numeri, testi, immagini, suoni) ma anche istruzioni (azioni da compiere sui dati). Ne consegue che il computer è una macchina programmabile, in grado di eseguire una quantità illimitata di elaborazioni diverse (anziché una quantità limitata come i tasti che corrispondono alle operazioni della calcolatrice). Dunque nel computer le istruzioni sono una categoria speciale di informazioni.

Elaborare le informazioni, che è l'attività propria del computer, significa ricevere dei dati e produrre dei risultati (informazioni). I dati ricevuti si chiamano **input**, i risultati prodotti si chiamano **output**. Un programma/algoritmo a questo punto si può definire come una sequenza di istruzioni che trasformano l'input nell'output.

L'utilizzatore che è in grado di introdurre nella macchina soltanto dati (affidandosi a elaborazioni predefinite) si chiama **utente**; l'utilizzatore che è in grado di introdurre anche istruzioni (e quindi di dire al computer non solo che cosa fare, ma anche come farlo) si chiama **programmatore**.

*Utente e programmatore sono due ruoli diversi e bisogna tenerne conto anche quando ad interpretarli è la stessa persona.*

## Il computer visto come l'essere umano

Si può paragonare il computer all'essere umano e quindi l'informatica (la scienza che studia il computer) alle scienze che studiano l'essere umano.

Le componenti del computer si suddividono in:

- **Hardware:** è il corrispettivo del corpo umano ed è la componente materiale del computer. Comprende tutto ciò che nel computer occupa una porzione di spazio fisico e quindi si può vedere e toccare
- **Software:** è invece il corrispettivo della mente intesa come pensieri, cioè la componente "ideale" del computer. Comprende tutte le informazioni che il computer elabora e tutte le istruzioni con cui le elabora

- **Periferiche di Input:** è il corrispettivo della percezione (i dati che riceviamo dal mondo)
- **Periferiche di Output:** è il corrispettivo dell'azione (i risultati che produciamo nel mondo)

*Il paragone fra computer e essere umano è utile per chiarire alcuni concetti, ma va trattato con l'opportuna cautela. La differenza fondamentale è che nel computer il software è separabile dall'hardware, mentre nell'essere umano – almeno allo stato attuale delle conoscenze scientifiche – la mente non è separabile dal corpo.*

## I componenti hardware di un computer

---

**Memoria centrale:** è lo spazio fisico in cui vengono immagazzinate le informazioni e le istruzioni che compongono i programmi. La memoria centrale viene suddivisa in due componenti:

- **RAM (Random Access Memory):** memoria “volatile” o memoria di lavoro. In essa vengono conservati(*mantenuti*) i dati in corso di elaborazione e le istruzioni del programma in esecuzione (perciò tutti i dati durante un'intera sessione di lavoro). Allo spegnimento del computer tutto ciò che è contenuto nella RAM viene perso (per questo viene definita volatile).
- **ROM (Read Only Memory):** la parte non volatile della memoria centrale, serve nella fase di accensione del computer (*bootstrap*). Contiene parti essenziali del software di sistema quali il *BIOS*.

**CPU (Central Processing Unit):** è il “cervello” del computer ed è composta da un'Unità di Controllo (CU) che governa il funzionamento della macchina e gestisce le relazioni con memoria e CPU e da un'Unità di Calcolo (ALU) che esegue operazioni aritmetiche e logiche.

**Periferiche di comunicazione:** si definiscono tali i dispositivi che permettono di svolgere le operazioni di *input* (tastiera, mouse) e di *output* (schermo, stampante). Il modem è una periferica di comunicazione che serve a comunicare, in ambo le direzioni(input/output), con un altro computer anziché con l'utente.

**Periferiche di memorizzazione(storage) o memorie secondarie:** si definiscono tali i dispositivi (hard-disk, chiavette, CD, DVD) che permettono di sopperire alle dimensioni limitate e alla natura “volatile” della memoria centrale.

## Cos'è un algoritmo

---

Sequenza **finita** e **ordinata** di passi/operazioni che portano alla realizzazione di uno specifico compito. L'algoritmo di un'operazione complessa può essere scomposto in una sequenza di istruzioni più semplici.

Esempi: calcoli matematici, massimo comune divisore, istruzioni di un elettrodomestico, prelievamento Bancomat, ricette di cucina.

*N.B. Un computer è un esecutore di algoritmi*

Proprietà di un algoritmo: **correttezza** (deve giungere alla soluzione del dato problema) ed **efficienza** (dare la soluzione nel modo più veloce, utilizzando la minima quantità di risorse fisiche).

Metodi di rappresentazione di un algoritmo:

- Linguaggio naturale
- Diagramma a blocchi
- Pseudo codice
- Linguaggio di programmazione

## Cos'è un programma

---

Insieme di uno o più algoritmi scritti in un determinato linguaggio di programmazione. Il processore del computer esegue i programmi passo-passo in modo preciso e veloce. L'insieme dei **dati** e delle **istruzioni** finalizzate alla produzione di un risultato o di uno scopo ben preciso si chiama programma.

## Gli algoritmi sono parametrici

---

- Producono un risultato(*informazione*) che dipende da un insieme di dati di partenza

- Descrivono la soluzione non di un singolo problema, ma di una intera classe di problemi strutturalmente equivalenti

*Esempi:*

- l'algoritmo per la moltiplicazione di due numeri specifica come effettuare il prodotto di tutte le possibili coppie di numeri
- l'algoritmo per la ricerca di un libro nello schedario della biblioteca vale per tutti i possibili libri

Le istruzioni dell'algoritmo fanno riferimento a **variabili**, il cui valore non è fissato a priori ma cambia a seconda della situazione elaborativa in cui il processore del computer si trova.

## Cos'è una variabile

---

È un *dato*, formato dal tipo, un'etichetta e un valore, mantenuta all'interno della memoria RAM.

Il **tipo** di una variabile definisce come interpretare i dati memorizzati e lo spazio necessario alla sua memorizzazione (ad esempio numerico o stringa).

L'**etichetta** è il nome con cui riferirsi alla variabile all'interno del linguaggio di programmazione. In generale i linguaggi di programmazione distinguono minuscolo e maiuscolo (**case-sensitive**). L'etichetta deve rispettare le seguenti regole sintattiche: deve iniziare con una lettera minuscola o il simbolo underscore ( `_` ) e può contenere lettere minuscole/maiuscole, cifre da 0 a 9 e il simbolo underscore ( `_` ).

Il **valore** che può assumere una variabile, in generale, dipende dal tipo della variabile.

Le variabili devono essere **dichiarate** prima di poter essere utilizzate, possono essere **inizializzate** all'atto della dichiarazione. All'atto della dichiarazione il tipo della variabile, se non è stato assegnato, è indefinito (**undefined**). L'inizializzazione assegna un valore ad una variabile e di conseguenza il tipo se non è stato definito precedentemente.

Javascript e PHP non permettono di definire i tipi di variabili, sarà quindi il valore che andremo ad inserire a definire il tipo, per tale motivazione Javascript e PHP vengono detti linguaggi debolmente tipizzati (non fortemente tipizzati).

## Uso delle variabili

---

- All'interno di espressioni (l'esecutore usa il valore contenuto nelle variabili per calcolare il risultato dell'espressione, per esempio  $op1 + op2 \times op3$  oppure  $op1 / op2 - op3$ , ...)
- In istruzioni di assegnamento (introdurre nel contenitore identificato dal nome della variabile il valore specificato a destra dell'assegnamento, per esempio  $r = 35$  (asigna 35 alla variabile il cui nome è  $r$ ),  $pi = 3,14$ , ...)
- In istruzioni di assegnamento combinate con espressioni (asigna a una variabile il risultato ottenuto dalla valutazione di un'espressione, per esempio in " $circ = 2 \times r \times pi$ " il risultato dell'espressione  $2 \times r \times pi$  viene calcolato utilizzando i valori contenuti nelle variabili  $r$  e  $pi$  e il risultato viene poi assegnato alla variabile  $circ$ ; la stessa variabile può comparire in entrambi i lati dell'istruzione di assegnamento, per esempio in " $k = k + 1$ " il valore contenuto in  $k$  viene utilizzato per trovare il valore dell'espressione  $k + 1$  che viene memorizzato come nuovo valore di  $k$ .)

## Assegnazione di valori a variabili

---

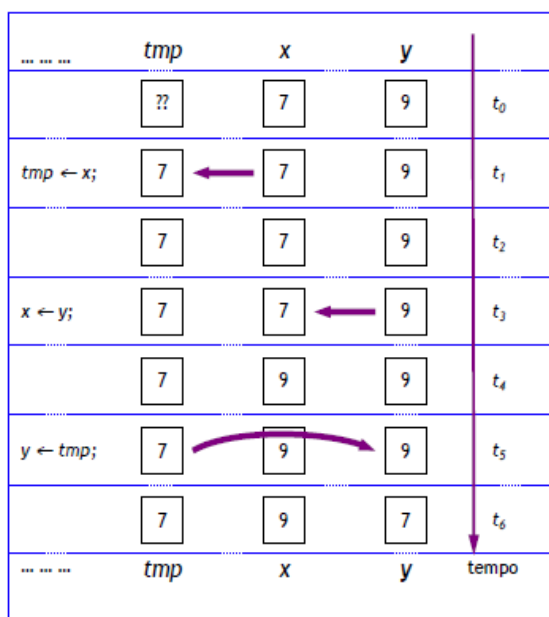
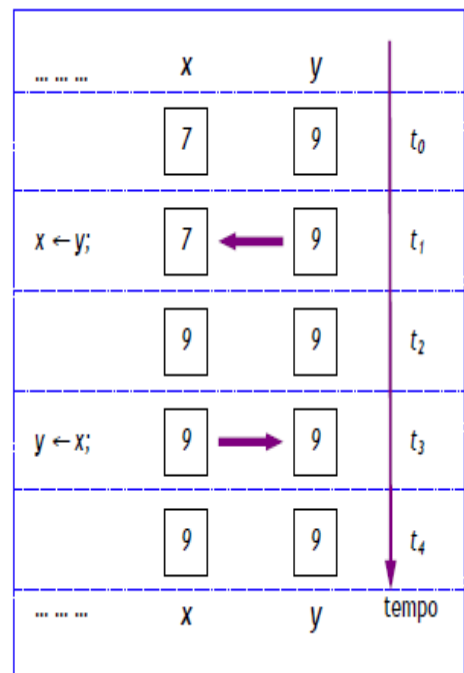
Il valore assegnato a una variabile si sostituisce a quello che era presente in precedenza, il vecchio valore non potrà più essere recuperato.

**Esempio:** si ipotizzi di voler scambiare i valori contenuti in due variabili  $x$  e  $y$ .

**Soluzione proposta:** doppio assegnamento del tipo

$x \leftarrow y$   
 $y \leftarrow x$

Per indicare che il valore di  $y$  deve essere copiato in  $x$  e che, nello stesso tempo, il valore di  $x$  sia trasferito in  $y$ . Le istruzioni però vengono eseguite in sequenza! Quindi l'assegnamento  $x \leftarrow y$  viene completato prima di iniziare  $y \leftarrow x$ .



**Soluzione corretta:** uso di una variabile aggiuntiva ( $tmp$ ), come strumento di memorizzazione temporanea (“buffer”) del valore originariamente contenuto in  $x$

$tmp \leftarrow x$   
 $x \leftarrow y$   
 $y \leftarrow tmp$

In questo modo lo scambio avviene senza perdere i valori originali.

## Dati e Istruzioni

- Tipi di dati
  - Numeri naturali o interi o reali (1, -2, 0.34)
  - Caratteri alfanumerici (A, B, a, c, 1, 2, -, |, ...)

- N.B. un insieme ordinato (l'ordine degli elementi ha importanza) di caratteri alfanumerici viene definita stringa (*il fatto che gli elementi dell'insieme abbiano un ordine non significa che gli elementi siano ordinati tra loro*).
- Dati logici o booleani (true, false)
- Array o vettore di n elementi ({1, 2, 3})
  - N.B. un array/vettore è un insieme ordinato (l'ordine degli elementi ha importanza) di elementi di un dato tipo (*il fatto che gli elementi dell'array abbiano un ordine non significa che gli elementi siano ordinati tra loro*). Essendo ordinato, è possibile accedere ad un elemento dell'array/vettore attraverso il suo indice, che corrisponde alla posizione all'interno dell'array/vettore.
- Istruzioni
  - Operazioni di Input/Output (es. leggi, scrivi)
  - Operazioni Aritmetico-logiche (es. somma =  $A + B$ )
  - Strutture di controllo (es. SE, RIPETI)
- Struttura dati di tipo
  - elementari (interi, alfanumerici, booleani, ...)
  - strutturati (array, matrici, ...)

N.B. Una **stringa** è un particolare array/vettore. Infatti è un insieme ordinato di singoli caratteri alfanumerici.

## Esempio di programma in linguaggio naturale: Determinare il maggiore tra due numeri

---

- P1 leggi un valore dall'esterno e assegnalo alla variabile **x**;
- P2 leggi un secondo valore dall'esterno e assegnalo alla variabile **y**;
- P3 calcola la differenza **d** fra **x** e **y**, cioè esegui  $d \leftarrow x - y$ ;
- P4 valuta se **d** è positivo: in caso *affermativo* prosegui con il passo P5, altrimenti (in caso *negativo*) salta al passo P7;
- P5 scrivi "il numero maggiore è " seguito dal valore di **x**;
- P6 salta al passo P11;
- P7 valuta se **d** è nullo: in caso *affermativo* prosegui con il passo P8, altrimenti (in caso *negativo*) salta al passo P10;
- P8 scrivi "i due numeri sono uguali";
- P9 salta al passo P11;

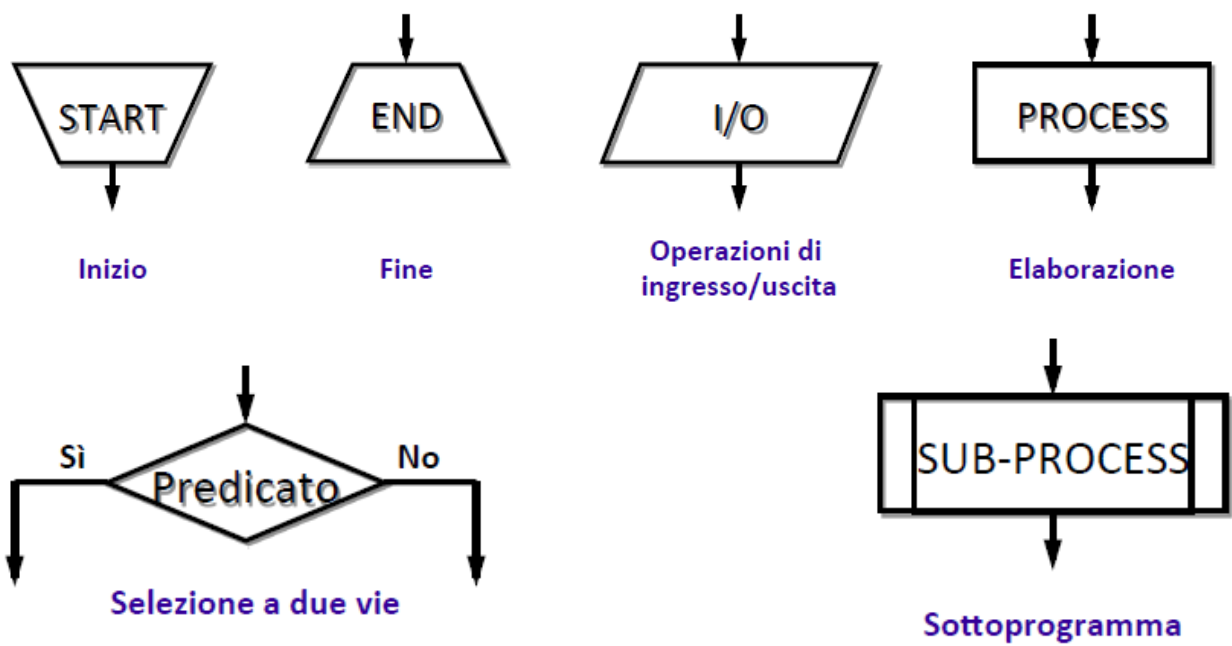


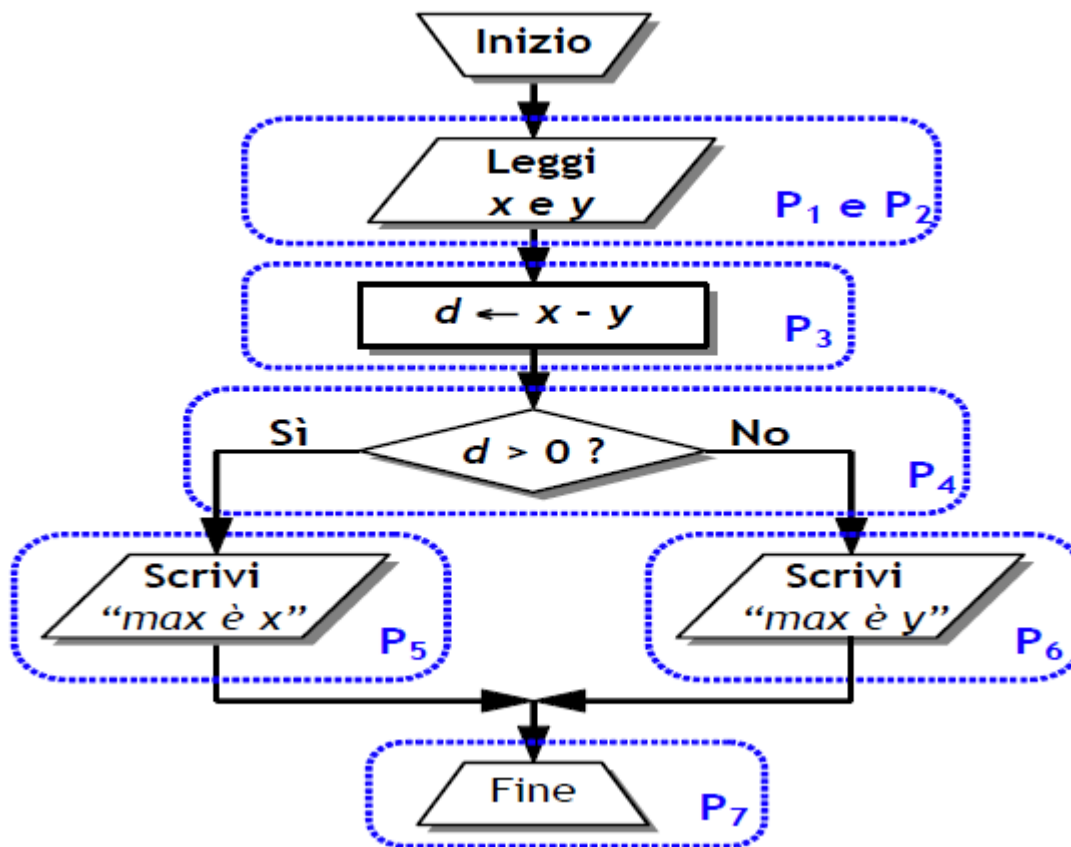
P10 scrivi “il numero maggiore è ” seguito dal valore di **y**;

P11 termina l’esecuzione.

## Struttura dei diagrammi di flusso

---





Leggi alfa, beta;

prodotto  $\leftarrow$  0;

Finché alfa > 0 ripeti

    prodotto  $\leftarrow$  prodotto + beta;

    alfa  $\leftarrow$  alfa - 1;

stampa prodotto;